Obtener la fecha y hora formateada en C.

Recopilado por Haylem Candelario Bauza. Documento Libre.
Origen. https://poesiabinaria.net/2012/06/obtener-la-fecha-y-hora-formateada-en-c/

Es un código muy simple, aunque tal vez si estáis empezando os resulte muy útil para que vuestros programas sean capaces de decir la hora o hacer alguna operación con la fecha

```
#include <time.h>
#include <stdio.h>

int main()
{
  time_t t;
  struct tm *tm;
  char fechayhora[100];

  t=time(NULL);
  tm=localtime(&t);
  strftime(fechayhora, 100, "%d/%m/%Y", tm);
  printf ("Hoy es: %s\n", fechayhora);
}
```

La forma de utilizar strftime() es pasando los siguientes parámetros:

Un array de char o char* donde almacenaremos la cadena de caracteres resultante
El tamaño máximo de caracteres que podemos ocupar, es decir, el tamaño reservado para la variable anterior
El formato en el que queremos presentar los datos, con algunas palabras clave que indicarán dónde va cada uno de los datos, los más importantes son:
%s : Número de segundos desde la época: 1 de Enero del 1970 a las 00:00
%d : Día del mes (del 01 al 31)
%m : Mes (del 01 al 12)
%Y : Año (con 4 cifras. Ej: 2012)
%y : Año (con 2 cifras. Ej : 12)
%H : Horas en formato 24h (de 00 a 23)
%I : Horas en formato 12h) (de 01 a 12)
%M : Minutos (de 00 a 59)
%S : Segundos (de 00 a 59)
%z : Huso horario (diferencia horaria con respecto a GMT)
%u : Día de la semana (del 1 al 7 donde 1 es Lunes)
%w : Día de la semana (del 0 al 6 donde 0 es Domingo)
Hay muchos más, para obtener la lista completa, utilizad: «$ man strftime» o «$ man date» en la línea de comandos
Puntero a struct tm de donde sacamos la información (debemos llamar siempre previamente a la función localtime()
Bueno, no todo queda ahí, podemos sacar la información directamente del (struct tm*), por ejemplo así:

```
#include <time.h>
#include <stdio.h>

int main()
{
  time_t t;
  struct tm *tm;
  char fechayhora[100];

  t=time(NULL);
  tm=localtime(&t);
  printf ("Hoy es: %02d/%02d/%d\n", tm->tm_mday, tm->tm_mon, 1900+tm->tm_year);
}
```

Es lo mismo de antes, pero con algunos detalles, por ejemplo el año, cuenta el número de años desde 1900, por lo que si queremos el año completo debemos sumar 1900 como en el ejemplo. Podemos ver la estructura struct tm aquí:

```
struct tm {
int tm_sec; /* segundos */
int tm_min; /* minutos */
int tm_hour; /* horas */
int tm_mday; /* día del mes */
int tm_mon; /* mes */
int tm_year; /* año */
int tm_wday; /* día de la semana */
int tm_yday; /* día del año */
int tm_isdst; /* cambio horario verano/invierno */
};
```

Fuente: man localtime

Aunque extraer estos valores directamente de struct tm puede ser incómodo, nos sirve para analizar los datos por separado y obtenerlos como valores enteros, por ejemplo para hacer cálculos. Un buen ejercicio es intentar traducir los nombres de los meses y día de la semana (aunque strftime(), comentada anteriormente es capaz de devolver esta información en el idioma del sistema, en ocasiones el idioma de nuestra aplicación no tiene por qué coincidir), podemos hacer lo siguiente:

```
include <time.h>
#include <stdio.h>

int main()
{
  time_t t;
  struct tm *tm;
  char fechayhora[100];
  char *meses[12]={"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"};

  t=time(NULL);
  tm=localtime(&t);
  printf ("Hoy es: %02d/%s/%d\n", tm->tm_mday, meses[tm->tm_mon], 1900+tm->tm_year);
}
```